Down the Borel Hierarchy: Solving Muller Games via Safety Games ☆

Daniel Neider[a], Roman Rabinovich[b,1], Martin Zimmermann[a,c,2]

[a] *Lehrstuhl für Informatik 7, RWTH Aachen University, 52056 Aachen, Germany*
[b] *Lehr- und Forschungsgebiet Mathematische Grundlagen der Informatik, RWTH Aachen University, 52056 Aachen, Germany*
[c] *Institute of Informatics, University of Warsaw, 02-097 Warsaw, Poland*

Abstract

We transform a Muller game with n vertices into a safety game with $(n!)^3$ vertices whose solution allows us to determine the winning regions of the Muller game and to compute a finite-state winning strategy for one player. This yields a novel antichain-based memory structure, a compositional solution algorithm, and a natural notion of permissive strategies for Muller games. Moreover, we generalize our construction by presenting a new type of game reduction from infinite games to safety games and show its applicability to several other winning conditions.

Keywords: Muller Games, Safety Games, Permissive Strategies, Game Reductions
2000 MSC: 68Q45, 68Q60

1. Introduction

Muller games are a source of interesting and challenging questions in the theory of infinite games. They are expressive enough to describe all ω-regular properties. Also, all winning conditions that depend only on the set of vertices visited infinitely often can trivially be reduced to Muller games. Hence, they subsume Büchi, co-Büchi, parity, Rabin, and Streett conditions. Furthermore, Muller games are not positionally determined, i.e., both players need memory

☆This work was supported by the projects *Games for Analysis and Synthesis of Interactive Computational Systems (GASICS)* and *Logic for Interaction (LINT)* of the *European Science Foundation* and by the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement 239850 (SOSNA).

Email addresses: neider@automata.rwth-aachen.de (Daniel Neider), roman.rabinovich@tu-berlin.de (Roman Rabinovich), zimmermann@react.uni-saarland.de (Martin Zimmermann)

[1] Present address: Logic and Semantics Research Group, Technische Universität Berlin, 10587 Berlin, Germany

[2] Present address: Reactive Systems Group, Saarland University, 66123 Saarbrücken, Germany

to implement their winning strategies. In this work, we consider three aspects of Muller games: solution algorithms, memory structures, and quality measures for strategies.

To date, there are two main approaches to solve Muller games: direct algorithms and reductions. Examples for the first approach are Zielonka's recursive polynomial space algorithm [1] which is based on earlier work by McNaughton [2], and Horn's polynomial time algorithm for explicit Muller games [3]. The second approach is to reduce a Muller game to a parity game using Zielonka trees [4] or latest appearance records (LAR) [5].

In general, the number of memory states needed to win a Muller game is prohibitively large [4]. Hence, a natural task is to reduce this number (if possible) and to find new memory structures which may implement small winning strategies in subclasses of Muller games.

As for the third aspect, to the best of our knowledge there is no previous work on quality measures for strategies in Muller games. This is in contrast to other winning conditions. Recently, much attention is being paid to not just synthesize some winning strategy, but to find an optimal one according to a certain quality measure, e.g., waiting times in request-response games [6] and their extensions [7], permissiveness in parity games [8, 9], bounds in finitary games [10] and games with costs [11], and the use of weighted automata in quantitative synthesis [12, 13].

Inspired by work of McNaughton [14], we present a framework to deal with all three issues. Our main contributions are a novel algorithm and a novel type of memory structure for Muller games. We also obtain a natural quality measure for strategies in Muller games and are able to extend the definition of permissiveness to Muller games.

While investigating the interest of Muller games for "casual living-room recreation" [14], McNaughton introduced scoring functions which describe the progress a player is making towards winning a play of the game: consider a Muller game $(\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$, where $\mathcal{A}$ is the arena and $(\mathcal{F}_0, \mathcal{F}_1)$ is a partition of the set of strongly connected subsets in $\mathcal{A}$ used to determine the winner. Then, the score of a set F of vertices measures how often F has been visited completely since the last visit of a vertex not in F. Player i wins a play in the Muller game if and only if there is an $F \in \mathcal{F}_i$ such that the score of F tends to infinity while being reset only finitely often (a reset occurs whenever a vertex outside F is visited).

McNaughton proved the existence of strategies for the winning player that bound her opponent's scores by $|\mathcal{A}|!$ [14], if the play starts in her winning region. The characterization above implies that such a strategy is necessarily winning. The bound $|\mathcal{A}|!$ was subsequently improved to two (and shown to be tight) [15]. Since some score eventually reaches value three, the winning regions of a Muller game can therefore be determined by solving the reachability game on a finite tree in which a player wins if she is the first to reach a score of three [3]. However,

[3] This reachability game was the object of McNaughton's study of games playable by hu-

it is cumbersome to obtain a winning strategy for the infinite-duration Muller game from a winning strategy for the finite-duration reachability game. The reason is that one has to carefully concatenate finite plays of the reachability game to an infinite play of the Muller game: reaching a score of three infinitely often does not prevent the opponent from visiting other vertices infinitely often.

Our Contributions

The ability to bound the losing player's scores can be seen as a safety condition as well. We use this to devise an algorithm to solve Muller games that computes both winning regions and a winning strategy for one player via a reduction to safety games. However, we do not obtain a winning strategy for the other player. In general, it is impossible to reduce a Muller game to a safety game whose solution yields winning strategies for both players, since safety conditions are on a lower level of the Borel hierarchy than Muller conditions.

Given a Muller game, we construct a safety game in which the scores of Player 1 are tracked (up to score three). Player 0 wins the safety game, if she can prevent Player 1 from ever reaching a score of three for every $F \in \mathcal{F}_1$. This allows us to compute the winning region of the Muller game by solving a safety game. Making use of the conjunctive nature of the winning condition of the safety game, we are also able to give a compositional algorithm for solving Muller games by solving the resulting safety game compositionally.

Furthermore, by exploiting the intrinsic structure of the safety game's arena we present an antichain-based memory structure for Muller games. Unlike the memory structures induced by Zielonka trees, which disregard the structure of the arena, and the ones induced by LARs, which disregard the structure of the winning condition $(\mathcal{F}_0, \mathcal{F}_1)$, our memory structure takes both directly into account: a simple arena or a simple winning condition should directly lead to a small memory. The other two memory structures only take one source of simplicity into account. Also, our memory implements the most general nondeterministic winning strategy among those that prevent the opponent from reaching a certain score in a Muller game. Thus, our framework allows us to extend the notion of permissiveness from positionally determined games to games that require memory.

Our idea of turning a Muller game into a safety game can be generalized to other types of winning conditions. We define a novel notion of reduction from infinite games to safety games which not only subsumes our construction but generalizes several constructions found in the literature. Based on work on small progress measures for solving parity games [16], Bernet, Janin, and Walukiewicz showed how to determine the winning regions in a parity game and a winning strategy for one player by reducing it to a safety game [8]. Furthermore, Schewe and Finkbeiner [17] as well as Filiot, Jin, and Raskin [18] used a translation from co-Büchi games to safety games in their work on bounded synthesis and LTL realizability, respectively. We present further examples and show that our

mans, which, for practical reasons, should end after a bounded number of steps.

reduction allows us to determine the winning region and a winning strategy for one player by solving a safety game. Thus, all these games can be solved by a new type of reduction and an algorithm for safety games. Our approach simplifies the winning condition of the game, even down the Borel hierarchy. However, this is offset by an increase in the size of the arena. Nevertheless, in the case of Muller games, our arena is only cubically larger than the arena constructed in the reduction to parity games. Furthermore, a safety game can be solved in linear time, while the question of whether there is a polynomial time algorithm for parity games is open.

2. Definitions

The power set of a set V is denoted by 2^V, and $\mathbb{N}$ denotes the set of non-negative integers. The prefix relation on words is denoted by $\sqsubseteq$. For $\rho \in V^\omega$ and $L \subseteq V^\omega$ we define $\mathrm{Pref}(\rho) = \{w \in V^* \mid w \sqsubseteq \rho\}$ and $\mathrm{Pref}(L) = \bigcup_{\rho \in L} \mathrm{Pref}(\rho)$. For $w = w_1 \cdots w_n \in V^+$, let $\mathrm{Last}(w) = w_n$.

An arena $\mathcal{A} = (V, V_0, V_1, E)$ consists of a finite, directed graph (V, E), $V_0 \subseteq V$ and $V_1 = V \setminus V_0$, where V_i denotes the positions of Player i (Player 0's vertices are drawn as ellipses, Player 1's as rectangles). We require every vertex to have an outgoing edge to avoid the nuisance of dealing with finite plays. The size $|\mathcal{A}|$ of $\mathcal{A}$ is the cardinality of V. A loop $C \subseteq V$ in $\mathcal{A}$ is a strongly connected subset of V, i.e., for every $v, v' \in C$ there is a path from v to v' that only visits vertices in C. A play in $\mathcal{A}$ starting in $v \in V$ is an infinite sequence $\rho = \rho_0\rho_1\rho_2 \ldots$ such that $\rho_0 = v$ and $(\rho_n, \rho_{n+1}) \in E$ for all $n \in \mathbb{N}$. The occurrence set $\mathrm{Occ}(\rho)$ and infinity set $\mathrm{Inf}(\rho)$ of ρ are given by $\mathrm{Occ}(\rho) = \{v \in V \mid \exists n \in \mathbb{N} \text{ such that } \rho_n = v\}$ and $\mathrm{Inf}(\rho) = \{v \in V \mid \exists^\omega n \in \mathbb{N} \text{ such that } \rho_n = v\}$. We also use the occurrence set of a finite play infix w, which is defined in the same way. The infinity set of a play is always a loop in the arena. A game $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ consists of an arena $\mathcal{A}$ and a set $\mathrm{Win} \subseteq V^\omega$ of winning plays for Player 0. The set of winning plays for Player 1 is $V^\omega \setminus \mathrm{Win}$.

A strategy for Player i is a mapping $\sigma\colon V^*V_i \to V$ such that $(v, \sigma(wv)) \in E$ for all $wv \in V^*V_i$. We say that σ is positional if $\sigma(wv) = \sigma(v)$ for every $wv \in V^*V_i$. Often, we denote positional strategies as mappings $\sigma\colon V_i \to V$. A play $\rho_0\rho_1\rho_2 \ldots$ is consistent with σ if $\rho_{n+1} = \sigma(\rho_0 \cdots \rho_n)$ for every n with $\rho_n \in V_i$. For $v \in V$ and a strategy σ, we define the behavior of σ from v by $\mathrm{Beh}(v, \sigma) = \{\rho \in V^\omega \mid \rho \text{ is a play that starts in } v \text{ and is consistent with } \sigma\}$ and $\mathrm{Beh}(W, \sigma) = \bigcup_{v \in W} \mathrm{Beh}(v, \sigma)$ for $W \subseteq V$.

A strategy σ for Player i is a winning strategy from a set of vertices $W \subseteq V$ if every $\rho \in \mathrm{Beh}(W, \sigma)$ is won by Player i. The winning region $W_i(\mathcal{G})$ of Player i in $\mathcal{G}$ is the set of vertices from which Player i has a winning strategy. We always have $W_0(\mathcal{G}) \cap W_1(\mathcal{G}) = \emptyset$ and $\mathcal{G}$ is determined if $W_0(\mathcal{G}) \cup W_1(\mathcal{G}) = V$. A winning strategy for Player i is uniform, if it is winning from $W_i(\mathcal{G})$.

A memory structure $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd})$ for an arena (V, V_0, V_1, E) consists of a finite set M of memory states, an initialization function $\mathrm{Init}\colon V \to M$, and an update function $\mathrm{Upd}\colon M \times V \to M$. The update function can be

extended to $\mathrm{Upd}^*\colon V^+ \to M$: $\mathrm{Upd}^*(\rho_0) = \mathrm{Init}(\rho_0)$ and $\mathrm{Upd}^*(\rho_0 \dots \rho_n\rho_{n+1}) = \mathrm{Upd}(\mathrm{Upd}^*(\rho_0 \dots \rho_n), \rho_{n+1})$. A next-move function (for Player i) $\mathrm{Nxt}\colon V_i \times M \to V$ has to satisfy $(v, \mathrm{Nxt}(v, m)) \in E$ for all $v \in V_i$ and $m \in M$. It induces a strategy σ for Player i with memory $\mathfrak{M}$ via $\sigma(\rho_0 \dots \rho_n) = \mathrm{Nxt}(\rho_n, \mathrm{Upd}^*(\rho_0 \dots \rho_n))$. The size of $\mathfrak{M}$ (and, slightly abusive, σ) is $|M|$. A strategy is finite-state if it can be implemented with a memory structure.

An arena $\mathcal{A} = (V, V_0, V_1, E)$ and a memory structure $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd})$ for $\mathcal{A}$ induce the expanded arena $\mathcal{A} \times \mathfrak{M} = (V \times M, V_0 \times M, V_1 \times M, E')$ where $((v, m), (v', m')) \in E'$ if and only if $(v, v') \in E$ and $\mathrm{Upd}(m, v') = m'$. Every play ρ in $\mathcal{A}$ has a unique extended play $\rho' = (\rho_0, m_0)(\rho_1, m_1)(\rho_2, m_2) \cdots$ in $\mathcal{A} \times \mathfrak{M}$ defined by $m_0 = \mathrm{Init}(\rho_0)$ and $m_{n+1} = \mathrm{Upd}(m_n, \rho_{n+1})$, i.e., $m_n = \mathrm{Upd}^*(\rho_0 \cdots \rho_n)$. A game $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ is reducible to $\mathcal{G}' = (\mathcal{A}', \mathrm{Win}')$ via $\mathfrak{M}$, written $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$, if $\mathcal{A}' = \mathcal{A} \times \mathfrak{M}$ and every play ρ in $\mathcal{G}$ is won by the player who wins the extended play ρ' in $\mathcal{G}'$, i.e., $\rho \in \mathrm{Win}$ if and only if $\rho' \in \mathrm{Win}'$.

Lemma 1. *Let $\mathcal{G}$ be a game with vertex set V and $W \subseteq V$. If $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$ and Player i has a positional winning strategy for $\mathcal{G}'$ from $\{(v, \mathrm{Init}(v)) \mid v \in W\}$, then she has a finite-state winning strategy with memory $\mathfrak{M}$ for $\mathcal{G}$ from W.*

We consider two types of games defined by specifying Win implicitly. A safety game is a tuple $\mathcal{G} = (\mathcal{A}, F)$ with $F \subseteq V$ and $\mathrm{Win} = \{\rho \in V^\omega \mid \mathrm{Occ}(\rho) \subseteq F\}$, i.e., Player 0 has to keep the play in F in order to win. Thus, vertices in F are called safe while vertices in $V \setminus F$ are called unsafe. In safety games, we allow unsafe terminal vertices, since Player 1 wins a play passing through such a vertex, i.e., there is no need to continue playing. A Muller game is a triple $\mathcal{G} = (\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$ where $\mathcal{F}_0$ is a set of loops of $\mathcal{A}$, $\mathcal{F}_1$ contains the loops which are not in $\mathcal{F}_0$, and $\mathrm{Win} = \{\rho \in V^\omega \mid \mathrm{Inf}(\rho) \in \mathcal{F}_0\}$, i.e., ρ is winning for Player i if and only if $\mathrm{Inf}(\rho) \in \mathcal{F}_i$. Safety games are determined with uniform positional strategies and Muller games are determined with uniform finite-state strategies of size $|\mathcal{A}|!$ [5].

3. Scoring Functions for Muller Games

In this section, we introduce scores and accumulators for Muller games. These concepts describe the progress of a player throughout a play. Intuitively, for each set $F \subseteq V$, the score of F of a play prefix w measures how often F has been visited completely since the last visit of a vertex that is not in F or since the beginning of w. The accumulator of the set F measures the progress made towards the next score increase: $\mathrm{Acc}_F(w)$ contains the vertices of F seen since the last increase of the score of F or the last visit of a vertex $v \notin F$, depending on which occurred later. For a more detailed treatment we refer to [14, 15].

Definition 1. *Let $w \in V^*$, $v \in V$, and $\emptyset \neq F \subseteq V$.*

- *Define $\mathrm{Sc}_F(\varepsilon) = 0$ and $\mathrm{Acc}_F(\varepsilon) = \emptyset$.*
- *If $v \notin F$, then $\mathrm{Sc}_F(wv) = 0$ and $\mathrm{Acc}_F(wv) = \emptyset$ (we say that Sc_F is reset).*

- *If* $v \in F$ *and* $\mathrm{Acc}_F(w) = F \setminus \{v\}$*, then* $\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w) + 1$ *and* $\mathrm{Acc}_F(wv) = \emptyset$ *(we say that* Sc_F *is increased).*
- *If* $v \in F$ *and* $\mathrm{Acc}_F(w) \neq F \setminus \{v\}$*, then* $\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w)$ *and* $\mathrm{Acc}_F(wv) = \mathrm{Acc}_F(w) \cup \{v\}$.

Also, for $\mathcal{F} \subseteq 2^V$ *define* $\mathrm{MaxSc}_{\mathcal{F}} \colon V^* \cup V^\omega \to \mathbb{N} \cup \{\infty\}$ *by* $\mathrm{MaxSc}_{\mathcal{F}}(\rho) = \max_{F \in \mathcal{F}} \sup_{w \sqsubseteq \rho} \mathrm{Sc}_F(w)$.

Example 1. *Let* $V = \{0, 1, 2\}$*,* $F = \{0, 1\}$*, and* $w = 10012100$*. We have* $\mathrm{Sc}_F(w) = 1$ *and* $\mathrm{Acc}_F(w) = \{0\}$*, but* $\mathrm{MaxSc}_{\{F\}}(w) = 2$*, due to the prefix* 1001*. The score for* F *is reset to* 0 *by the occurrence of* 2*, i.e.,* $\mathrm{Sc}_F(10012) = 0$ *and* $\mathrm{Acc}_F(10012) = \emptyset$.

If w is a play prefix with $\mathrm{Sc}_F(w) \geq 2$, then the set F is a loop of the arena. In an infinite play ρ, $\mathrm{Inf}(\rho)$ is the unique set F such that Sc_F tends to infinity while being reset to 0 only finitely often, since the score for every superset of $\mathrm{Inf}(\rho)$ is incremented only finitely often while the score for every other set (i.e., every non-superset) is reset infinitely often. Hence, $\mathrm{MaxSc}_{\mathcal{F}_{1-i}}(\rho) < \infty$ implies $\mathrm{Inf}(\rho) \in \mathcal{F}_i$. Also, we always have $\mathrm{Acc}_F(w) \subsetneq F$. Next, we give a score-based preorder and an induced equivalence relation on play prefixes.

Definition 2. *Let* $\mathcal{F} \subseteq 2^V$ *and* $w, w' \in V^+$.

1. *Define* $w \leq_{\mathcal{F}} w'$*, if* $\mathrm{Last}(w) = \mathrm{Last}(w')$ *and for all* $F \in \mathcal{F}$*:*
   - $\mathrm{Sc}_F(w) < \mathrm{Sc}_F(w')$*, or*
   - $\mathrm{Sc}_F(w) = \mathrm{Sc}_F(w')$ *and* $\mathrm{Acc}_F(w) \subseteq \mathrm{Acc}_F(w')$.
2. *Furthermore, define* $w =_{\mathcal{F}} w'$*, if* $w \leq_{\mathcal{F}} w'$ *and* $w' \leq_{\mathcal{F}} w$.

The condition $w =_{\mathcal{F}} w'$ is equivalent to $\mathrm{Last}(w) = \mathrm{Last}(w')$ and for every $F \in \mathcal{F}$ the equalities $\mathrm{Sc}_F(w) = \mathrm{Sc}_F(w')$ and $\mathrm{Acc}_F(w) = \mathrm{Acc}_F(w')$ hold. Thus, $=_{\mathcal{F}}$ is an equivalence relation. Both $\leq_{\mathcal{F}}$ and $=_{\mathcal{F}}$ are preserved under concatenation, i.e., $=_{\mathcal{F}}$ is a congruence.

Lemma 2. *Let* $\mathcal{F} \subseteq 2^V$ *and* $w, w' \in V^+$.

1. *If* $w \leq_{\mathcal{F}} w'$*, then* $wu \leq_{\mathcal{F}} w'u$ *for all* $u \in V^*$.
2. *If* $w =_{\mathcal{F}} w'$*, then* $wu =_{\mathcal{F}} w'u$ *for all* $u \in V^*$.

Proof. 1. It suffices to show that $w \leq_{\mathcal{F}} w'$ implies $wv \leq_{\mathcal{F}} w'v$ for all $v \in V$. So, let $F \in \mathcal{F}$: if $v \notin F$, then we have $\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w'v) = 0$ and $\mathrm{Acc}_F(wv) = \mathrm{Acc}_F(w'v) = \emptyset$, i.e., $wv =_{\mathcal{F}} w'v$.

Now, suppose we have $v \in F$. First, consider the case $\mathrm{Sc}_F(w) < \mathrm{Sc}_F(w')$: then, either the score of F does not increase in wv and we have

$$\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w) < \mathrm{Sc}_F(w') \leq \mathrm{Sc}_F(w'v)$$

or the score increases in wv and we have

$$\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w) + 1 \leq \mathrm{Sc}_F(w') \leq \mathrm{Sc}_F(w'v)$$

and $\mathrm{Acc}_F(wv) = \emptyset$, due to the score increase. Thus, $wv \leq_{\mathcal{F}} w'v$ in both cases.

Now, consider the case $\mathrm{Sc}_F(w) = \mathrm{Sc}_F(w')$ and $\mathrm{Acc}_F(w) \subseteq \mathrm{Acc}_F(w')$. If $\mathrm{Acc}_F(w) = F \setminus \{v\}$, then also $\mathrm{Acc}_F(w') = F \setminus \{v\}$, as the accumulator for F can never be F. In this situation, we have

$$\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w) + 1 = \mathrm{Sc}_F(w') + 1 = \mathrm{Sc}_F(w'v)$$

and $\mathrm{Acc}_F(wv) = \mathrm{Acc}_F(w'v) = \emptyset$. Otherwise, we have

$$\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w) = \mathrm{Sc}_F(w') \leq \mathrm{Sc}_F(w'v) \ .$$

If $\mathrm{Sc}_F(w') < \mathrm{Sc}_F(w'v)$, then we are done. So, consider the case $\mathrm{Sc}_F(w') = \mathrm{Sc}_F(w'v)$: we have

$$\mathrm{Acc}_F(wv) = \mathrm{Acc}_F(w) \cup \{v\} \subseteq \mathrm{Acc}_F(w') \cup \{v\} = \mathrm{Acc}_F(w'v) \ ,$$

due to $\mathrm{Acc}_F(w) \subseteq \mathrm{Acc}_F(w')$ and the fact that the score for $w'v$ does not increase, which implies that the accumulator for $w'v$ is obtained by adding v to the accumulator of w'. This completes the proof.

2. Apply 1. to the definition of $=_{\mathcal{F}}$. □

4. Solving Muller Games by Solving Safety Games

In this section, we show how to solve a Muller game by solving a safety game. Our approach is based on the existence of winning strategies for Muller games that bound the losing player's scores by two.

Lemma 3 ([15]). *In every Muller game $\mathcal{G} = (\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$ Player i has a uniform winning strategy σ such that $\mathrm{MaxSc}_{\mathcal{F}_{1-i}}(\rho) \leq 2$ for every $\rho \in \mathrm{Beh}(W_i(\mathcal{G}), \sigma)$.*

The following example shows that the bound two is tight.

Example 2. *Consider the Muller game $\mathcal{G} = (\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$, where $\mathcal{A}$ is depicted in Figure 1, $\mathcal{F}_0 = \{\{0\}, \{2\}, \{0,1,2\}\}$, and $\mathcal{F}_1 = \{\{0,1\}, \{1,2\}\}$. By alternatingly moving to 0 and to 2, Player 0 wins from every vertex, and she bounds Player 1's scores by two. However, he is able to achieve a score of two: consider a play starting at 1 and suppose (w.l.o.g.) that Player 0 moves to vertex 0. Then, Player 1 uses the self-loop once before moving back to 1, thereby reaching a score of two for the loop $\{0,1\} \in \mathcal{F}_1$.*

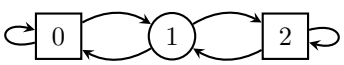

Figure 1: The arena $\mathcal{A}$ for Example 2.

A simple consequence of Lemma 3 is that a vertex v is in Player 0's winning region of the Muller game $\mathcal{G}$ if and only if she can prevent her opponent from ever reaching a score of three for a set in $\mathcal{F}_1$. This is a safety condition which only talks about small scores of one player. To determine the winning regions of $\mathcal{G}$, we construct an arena that keeps track of the scores of Player 1 up to threshold three. The winning condition F of the safety game requires Player 0 to prevent a score of three for her opponent. This idea is formalized in the proof of our main theorem, which we present in the remainder of this section.

Theorem 1. *Let $\mathcal{G}$ be a Muller game with vertex set V. One can effectively construct a safety game $\mathcal{G}_S$ with vertex set V^S and a mapping $f\colon V \to V^S$ with the following properties:*

1. *For every $v \in V$: $v \in W_i(\mathcal{G})$ if and only if $f(v) \in W_i(\mathcal{G}_S)$.*
2. *Player 0 has a uniform finite-state winning strategy for $\mathcal{G}$ with memory states $M \subseteq W_0(\mathcal{G}_S)$.*
3. *$|V^S| \leq (|V|!)^3$.*

Note that the first statement refers to both players while the second one only refers to Player 0. This is due to the fact that the safety game keeps track of Player 1's scores only, which allows Player 0 to prove that she can prevent him from reaching a score of three. But as soon as a score of three is reached, the play is stopped. To obtain a winning strategy for Player 1, we have to swap the roles of the players and construct a safety game which keeps track of the scores of Player 0. Alternatively, we could construct an arena which keeps track of both player's scores. However, that would require us to define two safety games in this arena: one in which Player 0 has to avoid a score of three for Player 1 and vice versa. This arena is larger than the ones in which only the scores of one player are tracked (but still smaller than $(|V|!)^3$). It is well-known that it is impossible to reduce a Muller game to a single safety game and thereby obtain winning strategies for both players. We come back to this in Section 7.

We begin the proof of Theorem 1 by defining the safety game $\mathcal{G}_S$. Let $\mathcal{G} = (\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$ with arena $\mathcal{A} = (V, V_0, V_1, E)$. We define

$$\text{Plays}_{<3} = \{w \mid w \text{ play prefix in } \mathcal{G} \text{ and } \text{MaxSc}_{\mathcal{F}_1}(w) < 3\}$$

to be the set of play prefixes in $\mathcal{G}$ in which the scores of Player 1 are at most two and we define

$$\begin{aligned}\text{Plays}_{=3} = \{w_0 \cdots w_{n+1} \mid\ & w_0 \cdots w_{n+1} \text{ play prefix in } \mathcal{G},\\ & \text{MaxSc}_{\mathcal{F}_1}(w_0 \cdots w_n) \leq 2, \text{ and } \text{MaxSc}_{\mathcal{F}_1}(w_0 \cdots w_n w_{n+1}) = 3\}\end{aligned}$$

to be the set of play prefixes in which Player 1 just reached a score of three. Furthermore, let $\text{Plays}_{\leq 3} = \text{Plays}_{<3} \cup \text{Plays}_{=3}$. Note that these definitions ignore the scores of Player 0. The arena of the safety game we are about to define is the $=_{\mathcal{F}_1}$-quotient of the unraveling of $\mathcal{A}$ up to the positions where Player 1 reaches a score of three for the first time (if he does at all).

Formally, we define $\mathcal{G}_S = ((V^S, V_0^S, V_1^S, E^S), F)$ where

- $V^S = \mathrm{Plays}_{\leq 3} /_{=_{\mathcal{F}_1}}$,
- $V_i^S = \{[w]_{=_{\mathcal{F}_1}} \mid [w]_{=_{\mathcal{F}_1}} \in V^S \text{ and } \mathrm{Last}(w) \in V_i\}$ for $i \in \{0,1\}$,
- $([w]_{=_{\mathcal{F}_1}}, [wv]_{=_{\mathcal{F}_1}}) \in E^S$ for every $w \in \mathrm{Plays}_{<3}$ and every v such that $(\mathrm{Last}(w), v) \in E$ (thus, every vertex in $\mathrm{Plays}_{=3} /_{=_{\mathcal{F}_1}}$ is terminal[4]), and
- $F = \mathrm{Plays}_{<3} /_{=_{\mathcal{F}_1}}$.

The definitions of V_0^S and V_1^S are independent of representatives, since $w =_{\mathcal{F}_1} w'$ implies $\mathrm{Last}(w) = \mathrm{Last}(w')$. Also, we have $V^S = V_0^S \cup V_1^S$ due to $V = V_0 \cup V_1$, and F is well-defined, since every equivalence class in $\mathrm{Plays}_{<3} /_{=_{\mathcal{F}_1}}$ is also one in $\mathrm{Plays}_{\leq 3} /_{=_{\mathcal{F}_1}}$. Finally, let $f(v) = [v]_{=_{\mathcal{F}_1}}$ for every $v \in V$.

Remark 1. *If* $([w]_{=_{\mathcal{F}_1}}, [w']_{=_{\mathcal{F}_1}}) \in E^S$*, then we have* $(\mathrm{Last}(w), \mathrm{Last}(w')) \in E$.

For the sake of readability, we denote $=_{\mathcal{F}_1}$-equivalence classes by $[w]$ from now on. All definitions and statements below are independent of representatives and we refrain from mentioning it from now on.

Example 3. *To illustrate the definitions, Figure 2 depicts the safety game* $\mathcal{G}_S$ *for the Muller game* $\mathcal{G}$ *from Example 2. The vertices* $[v]$ *for* $v \in V$ *are in the winning region of Player* 0*. This corresponds to the fact that Player* 0*'s winning region in the Muller game contains every vertex.*

The proof of Theorem 1 is split into several lemmata. Due to determinacy of both games, it suffices to consider only one player (we pick $i = 0$) to prove Theorem 1.1. To win the safety game, we simulate a winning strategy for the Muller game that bounds Player 1's scores by two. This suffices to avoid the vertices in $V^S \setminus F$, which encode that a score of three is reached.

Lemma 4. *For every* $v \in V$*: if* $v \in W_0(\mathcal{G})$*, then* $[v] \in W_0(\mathcal{G}_S)$.

Proof. Let σ be a uniform winning strategy for Player 0 for $\mathcal{G}$ that satisfies $\mathrm{MaxSc}_{\mathcal{F}_1}(\rho) \leq 2$ for every play $\rho \in \mathrm{Beh}(W_0(\mathcal{G}), \sigma)$. Due to Remark 1, every play prefix $[w_1] \cdots [w_n]$ in $\mathcal{G}_S$ can be mapped to a play $p([w_1] \cdots [w_n]) = \mathrm{Last}(w_1) \cdots \mathrm{Last}(w_n)$ in $\mathcal{G}$. We use this to define a strategy σ' for $\mathcal{G}_S$ by $\sigma'([w_1] \cdots [w_n]) = [w_n \cdot \sigma(p([w_1] \cdots [w_n]))]$ for every play prefix $[w_1] \cdots [w_n]$ of $\mathcal{G}_S$ with $[w_n] \in V_0^S \setminus F$. This is a legal move due to the definition of E^S and the restriction to plays ending in $V_0^S \setminus F$ is sufficient, since all other plays are already losing for Player 0. A simple induction shows $[w_1] \cdots [w_n]$ being consistent with σ' implies $p([w_1] \cdots [w_n])$ being consistent with σ.

It remains to show that σ' is winning for Player 0 from $\{[v] \mid v \in W_0(\mathcal{G})\}$. So, suppose σ' is not winning from some vertex $[v]$ with $v \in W_0(\mathcal{G})$, i.e., there

[4] This contradicts our requirements on an arena. However, every play visiting these vertices is losing for Player 0 no matter how it is continued. To simplify the following proofs, we refrain from defining outgoing edges for these vertices.

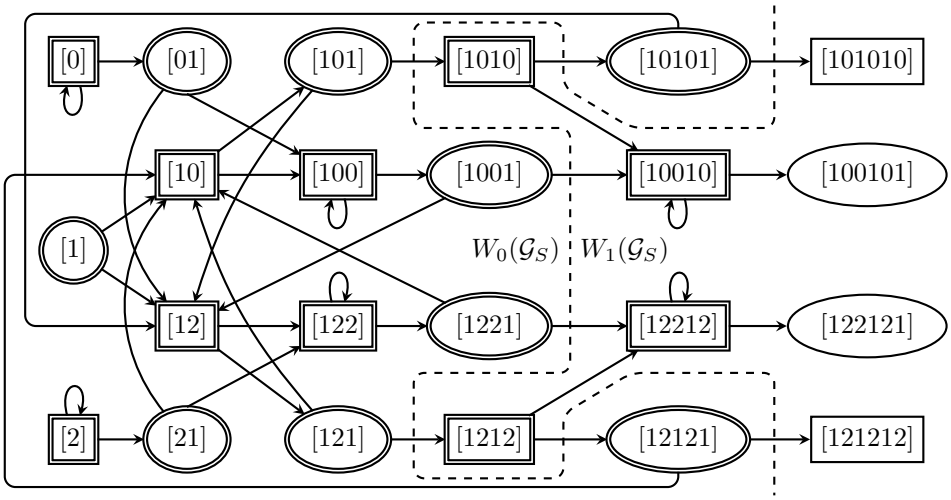

Figure 2: The safety game $\mathcal{G}_S$ for $\mathcal{G}$ from Example 2 (vertices drawn with double lines are in F); the dashed line separates the winning regions.

exists a play prefix $[w_1] \cdots [w_n]$ that is consistent with σ' such that $w_1 =_{\mathcal{F}_1} v$ and $w_n \in \text{Plays}_{=3}$. Another simple induction shows $p([w_1] \cdots [w_n]) \in [w_n]$. However, this contradicts our assumption on σ, since $p([w_1] \cdots [w_n])$ is consistent with σ, but allows Player 1 to reach a score of three. □

For the other direction of Theorem 1.1 we show that a subset of $W_0(\mathcal{G}_S)$ can be turned into a memory structure for Player 0 in the Muller game that induces a winning strategy. We use the $=_{\mathcal{F}_1}$-equivalence class of w as memory state to keep track of Player 1's scores in $\mathcal{G}$. However, instead of keeping track of the exact scores, it suffices to over-approximate them, which yields a smaller memory structure: instead of using all equivalence classes in the winning region of Player 0, we only use the maximal ones with respect to $\leq_{\mathcal{F}_1}$ that are reachable via a fixed positional winning strategy for her in the safety game. Formally, we have to lift $\leq_{\mathcal{F}_1}$ to equivalence classes by defining $[w] \leq_{\mathcal{F}_1} [w']$ if and only if $w \leq_{\mathcal{F}_1} w'$.

The following proof is similar to the reductions from co-Büchi [17, 18, 19] and parity games [8] to safety games, but for the more general case of Muller games. We come back to the similarities to those reductions when we determine permissive strategies in Section 6.

Lemma 5. *For all $v \in V$: if $[v] \in W_0(\mathcal{G}_S)$, then $v \in W_0(\mathcal{G})$.*

Proof. Let σ be a uniform positional winning strategy for Player 0 in $\mathcal{G}_S$ and let $R \subseteq V^S$ be the set of vertices reachable from $W_0(\mathcal{G}_S) \cap \{[v] \mid v \in V\}$ by plays consistent with σ. Every $[w] \in R \cap V_0^S$ has exactly one successor in R (of the form $[wv]$ for some $v \in V$) and dually, every successor of $[w] \in R \cap V_1^S$ (which

are exactly the classes $[wv]$ with $(\mathrm{Last}(w), v) \in E$) is in R. Now, let $R_{\max}$ be the set of $\leq_{\mathcal{F}_1}$-maximal elements of R. Applying the facts about successors of vertices in R stated above, we obtain the following remark.

Remark 2. *Let $R_{\max}$ be defined as above.*

1. *For every $[w] \in R_{\max} \cap V_0^S$, there is a $v \in V$ with $(\mathrm{Last}(w), v) \in E$ and there is a $[w'] \in R_{\max}$ such that $[wv] \leq_{\mathcal{F}_1} [w']$.*
2. *For every $[w] \in R_{\max} \cap V_1^S$ and each of its successors $[wv]$, there is a $[w'] \in R_{\max}$ such that $[wv] \leq_{\mathcal{F}_1} [w']$.*

Thus, instead of updating the memory from $[w]$ to $[wv]$ (i.e., keeping track of the exact scores) when processing a vertex v, we directly update it to a maximal element that is $\mathcal{F}_1$-larger than $[wv]$ (i.e., we over-approximate the exact scores). Formally, we define $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd})$ by $M = R_{\max} \cup \{\bot\}$ [5],

$$\mathrm{Init}(v) = \begin{cases} [w] & \text{if } [v] \in W_0(\mathcal{G}_S), \text{ for some } [w] \in R_{\max} \text{ with } [v] \leq_{\mathcal{F}_1} [w], \\ \bot & \text{otherwise,} \end{cases}$$

and

$$\mathrm{Upd}([w], v) = \begin{cases} [w'] & \text{if there exists } [w'] \in R_{\max} \text{ such that } [wv] \leq_{\mathcal{F}_1} [w'], \\ \bot & \text{otherwise.} \end{cases}$$

This implies $[w] \leq_{\mathcal{F}_1} \mathrm{Upd}^*(w)$ for every $w \in V^+$ with $\mathrm{Upd}^*(w) \neq \bot$. Thus, $\mathrm{Last}(w) = \mathrm{Last}(w')$, where $[w'] = \mathrm{Upd}^*(w)$. Using Remark 2, we define

$$\mathrm{Nxt}(v, [w]) = \begin{cases} v' & \text{if } \mathrm{Last}(w) = v, (v, v') \in E, \text{ and there exists } [w'] \in R_{\max} \\ & \quad \text{such that } [wv'] \leq_{\mathcal{F}_1} [w'], \\ v'' & \text{otherwise (where } v'' \text{ is some vertex with } (v, v'') \in E), \end{cases}$$

and $\mathrm{Nxt}(v, \bot) = v''$ for some v'' with $(v, v'') \in E$. The second case is just to match the formal definition of a next-move function; it is never invoked due to $\mathrm{Last}(w) = \mathrm{Last}(w')$ for $\mathrm{Upd}^*(w) = [w']$ or $\mathrm{Upd}^*(w) = \bot$.

It remains to show that the strategy σ implemented by $\mathfrak{M}$ and Nxt is a winning strategy for Player 0 from $W = \{v \mid [v] \in W_0(\mathcal{G}_S)\}$. An inductive application of Remark 2 shows that every play w that starts in W and is consistent with σ satisfies $\mathrm{Upd}^*(w) \neq \bot$. This bounds the scores of Player 1 by two, as we have $[w] \leq_{\mathcal{F}_1} \mathrm{Upd}^*(w) \in R_{\max} \subseteq \mathrm{Plays}_{<3}$ for every such play. Hence, σ is indeed a winning strategy for Player 0 from W. □

This construction and Lemma 4 complete the proof of Theorem 1.1 and imply Theorem 1.2.

[5] We use the memory state $\bot$ to simplify our proof. It is not reachable via plays that are consistent with the implemented strategy and can therefore be eliminated.

Corollary 1. *Player* 0 *has a uniform finite-state winning strategy whose memory states form an* $\leq_{\mathcal{F}_1}$*-antichain in* $W_0(\mathcal{G}_S)$*.*

To finish the proof of Theorem 1, we determine the size of $\mathcal{G}_S$ to prove the third statement. To this end, we use the concept of a *latest appearance record* (LAR) [2, 5]. Note that we do not need a hit position for our purposes.

A word $\ell \in V^+$ is an LAR if every vertex $v \in V$ appears at most once in ℓ. We map each $w \in V^+$ to a unique LAR, denoted by $\mathrm{LAR}(w)$, as follows: $\mathrm{LAR}(v) = v$ for every $v \in V$ and for $w \in V^+$ and $v \in V$ we define $\mathrm{LAR}(wv) = \mathrm{LAR}(w)v$ if $v \notin \mathrm{Occ}(w)$ and $\mathrm{LAR}(wv) = p_1p_2v$ if $\mathrm{LAR}(w) = p_1vp_2$. A simple induction shows that $\mathrm{LAR}(w)$ is indeed an LAR, which also ensures that the decomposition of w in the second case of the inductive definition is unique. We continue by showing that $\mathrm{LAR}(w)$ determines all but $|\mathrm{LAR}(w)|$ many of w's scores and accumulators.

Lemma 6. *Let* $w \in V^+$ *and* $\mathrm{LAR}(w) = v_k v_{k-1} \cdots v_1$*.*

1. *We can decompose* w *as* $w = x_k v_k x_{k-1} v_{k-1} \cdots x_2 v_2 x_1 v_1$ *where, for every* j*,* $x_j \in V^*$ *with* $\mathrm{Occ}(x_j) \subseteq \{v_1, \ldots, v_j\}$*.*
2. $\mathrm{Sc}_F(w) > 0$ *if and only if* $F = \{v_1, \ldots, v_j\}$ *for some* j*.*
3. *If* $\mathrm{Sc}_F(w) = 0$*, then* $\mathrm{Acc}_F(w) = \{v_1, \ldots, v_j\}$ *for the maximal* j *such that* $\{v_1, \ldots, v_j\} \subseteq F$ *and* $\mathrm{Acc}_F(w) = \emptyset$ *if no such* j *exists.*
4. *If* $\mathrm{Sc}_F(w) > 0$ *and* $F = \{v_1, \ldots, v_j\}$*, then* $\mathrm{Acc}_F(w) \in \{\emptyset\} \cup \{\{v_1, \ldots, v_{j'}\} \mid j' < j\}$*.*

Proof. 1. We prove the statement by induction over $|w|$. If $|w| = 1$, then the claim follows immediately from $w = \mathrm{LAR}(w)$. Now, let $|wv| > 1$. If $v \notin \mathrm{Occ}(w)$, then $\mathrm{LAR}(wv) = \mathrm{LAR}(w)v$ and the claim follows by induction hypothesis.

Now, suppose $\mathrm{LAR}(w) = p_1vp_2$ with $p_1 = v_k \cdots v_{i+1}$ and $p_2 = v_{i-1} \cdots v_1$, and hence $v_i = v$. By induction hypothesis, there exists a decomposition $w = x_k v_k x_{k-1} v_{k-1} \cdots v_2 x_1 v_1$ for some $x_j \in V^*$ such that for every j, $\mathrm{Occ}(x_j) \subseteq \{v_1, \ldots, v_j\}$. Also, we have $\mathrm{LAR}(wv) = p_1p_2v = v'_k \cdots v'_1$ where $v'_1 = v_i$, $v'_j = v_{j-1}$ for every $j \in \{2, \ldots, i\}$, and $v'_j = v_j$ for every $j \in \{i+1, \ldots, k\}$. Define $x'_1 = \varepsilon$, $x'_j = x_{j-1}$ for every $j \in \{2, \ldots, i-1\}$, $x'_i = x_i v_i x_{i-1}$, and $x'_j = x_j$ for every $j \in \{i+1, \ldots, k\}$. It is easy to verify that the decomposition $wv = x'_k v'_k x'_{k-1} v'_{k-1} \cdots v'_2 x'_1 v'_1$ has the desired properties.

2. We have $\mathrm{Sc}_F(w) > 0$ if and only if there exists a suffix x of w with $\mathrm{Occ}(x) = F$. Due to the decomposition characterization, having a suffix x with $\mathrm{Occ}(x) = F$ is equivalent to $F = \{v_1, \ldots, v_j\}$ for some j.

3. We have $\mathrm{Acc}_F(w) = \mathrm{Occ}(x)$ where x is the longest suffix of w such that the score of F does not change throughout x and $\mathrm{Occ}(x) \subseteq F$. Consider the decomposition characterization of w as above. We have $\{v_1, \ldots, v_j\} \subseteq \mathrm{Acc}_F(w)$, since $x = x_j v_j \cdots v_1 v_1$ is a suffix of w satisfying $\mathrm{Occ}(x) \subseteq F$. Furthermore, since $v_{j+1} \notin F$ by the maximality of j, this is the longest such suffix and we indeed have $\mathrm{Acc}_F(w) = \{v_1, \ldots, v_j\}$, since $\mathrm{Occ}(x)$ is a strict subset of F.

4. The latest increase of $\mathrm{Sc}_F(w)$ occurs after or at the last visit of v_j, since $\mathrm{Occ}(v_j x_{j-1} \cdots x_1 v_1) = F$. Hence, $\mathrm{Acc}_F(w)$ is the occurrence set of a suffix of $x_{j-1} \cdots x_1 v_1$ and the decomposition characterization yields the result. □

This characterization allows us to prove Theorem 1.3.

Lemma 7. *We have* $|V^S| \leq \left(\sum_{k=1}^{|V|} \binom{|V|}{k} \cdot k! \cdot 2^k \cdot k!\right) + 1 \leq (|V|!)^3$.

Proof. In every safety game, we can merge the vertices in $V \setminus F$ to a single vertex without changing $W_0(\mathcal{G})$. Since $[v] \in F$ for every v, we also retain the equivalence $v \in W_i(\mathcal{G}) \Leftrightarrow [v] \in W_i(\mathcal{G}_S)$.

Hence, it remains to bound the index of $\mathrm{Plays}_{<3} /_{=_{\mathcal{F}_1}}$. Lemma 6 shows that a play prefix $w \in V^+$ has $|\mathrm{LAR}(w)|$ many sets with non-zero score. Furthermore, the accumulator of the sets with score 0 is determined by $\mathrm{LAR}(w)$. Now, consider a play $w \in \mathrm{Plays}_{<3}$ and a set $F \in \mathcal{F}_1$ with non-zero score. We have $\mathrm{Sc}_F(w) \in \{1, 2\}$ and there are exactly $|F|$ possible values for $\mathrm{Acc}_F(w)$ due to Lemma 6.4. Finally, $\mathrm{LAR}(w) = \mathrm{LAR}(w')$ implies $\mathrm{Last}(w) = \mathrm{Last}(w')$. Hence, the index of $\mathrm{Plays}_{<3} /_{=_{\mathcal{F}_1}}$ is bounded by the number of LARs, which is $\sum_{k=1}^{n} \binom{n}{k} \cdot k!$, times the number of possible score and accumulator combinations for each LAR ℓ of length k, which is bounded by $2^k \cdot k!$. □

In the proof of Theorem 1.2, we used the maximal elements of Player 0's winning region of the safety game that are reachable via a fixed winning strategy. It is the choice of this strategy that determines the size of our memory structure, which could be much smaller than the rough upper bound stated in Lemma 7. However, finding a winning strategy for the safety game that visits at most $k \in \mathbb{N}$ vertices in an arena (from a fixed initial vertex) for a given k is NP-complete. This can be shown by a reduction from the vertex cover problem (see, e.g., [20] where a more general result is shown). Moreover, it is not even clear that a small strategy also yields few maximal elements.

In general, a player cannot prevent her opponent from reaching a score of two, but there are arenas in which she can do so. By first constructing the subgame $\mathcal{G}'_S$ up to threshold two (which is smaller than $\mathcal{G}_S$), we can possibly determine a subset of Player 0's winning region faster and obtain a (potentially) smaller finite-state winning strategy for this subset. But Example 2 shows that this approach is not complete.

5. Solving Muller Games Compositionally

In this section, we show how to solve Muller games compositionally. Our algorithm is again based on the reduction to safety games using scores. Remember that the winning condition of these games is conjunctive: for every $F \in \mathcal{F}_1$, Player 1 does not reach a score of three for F. The drawback of the monolithic algorithm in Section 4 is that the arena of $\mathcal{G}_S$ can be quite large. To cope with this, we use an algorithm that divides such a conjunctive winning condition into smaller games whose solutions allow us to solve the original game. Our compositional algorithm can reduce the memory requirements, but needs to solve a sequence of (smaller) safety games instead of one (large) game.

The general idea is as follows. Let $\mathcal{G} = (\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$ be a Muller game and $(\mathcal{F}_1^1, \mathcal{F}_1^2)$ a partition of $\mathcal{F}_1$ into non-empty sets. Instead of constructing the

whole safety game $\mathcal{G}_S$ as in Section 4, we solve two restricted safety games $\mathcal{G}_S^1$ and $\mathcal{G}_S^2$ where Player 0 has to avoid a score of three only for sets from $\mathcal{F}_1^1$ or $\mathcal{F}_1^2$, respectively. The restricted games are smaller than the original game $\mathcal{G}_S$, since they need to track fewer scores. By solving the restricted games, we obtain two winning regions W_1^i (for $i \in \{1, 2\}$) that contain all vertices from which Player 1 can enforce a score of three for sets in $\mathcal{F}_1^i$. This information can be used to identify vertices than can be removed from the original safety game $\mathcal{G}_S$ since Player 1 can win from these vertices already by considering only the set $\mathcal{F}_1^1$ or only the set $\mathcal{F}_1^2$. Thus, by solving two smaller safety games, we are able to reduce the size of $\mathcal{G}_S$ by identifying parts of Player 1's winning region before even solving $\mathcal{G}_S$. This process can then be recursively repeated by splitting $\mathcal{F}_1^1$ and $\mathcal{F}_1^2$ until we have reached singletons. Note however that it is neither a priori evident that this reduction is significant nor in which order the subsets from $\mathcal{F}_1$ should be considered. In the worst case, the compositional algorithm also needs to construct and solve the whole game $\mathcal{G}_S$. We discuss the advantages and disadvantages of the compositional algorithm at the end of this section.

For technical reasons, we move to a slightly different setting. We first show in Subsection 5.1 how to compositionally solve safety games whose set of vertices is a product of an arena $\mathcal{A}$ and memory structures $\mathfrak{M}_1, \ldots, \mathfrak{M}_k$ and whose winning condition depends on the memory states reached in the structures $\mathfrak{M}_j$. Intuitively, $\mathfrak{M}_j$ is used as "watchdog" for the scores of a set $F_j \in \mathcal{F}_1$ that signals if a score of three has been reached. Then, in Subsection 5.2, we provide memory structures such that their composition with the arena of the Muller game yields a safety game that is isomorphic to the game $\mathcal{G}_S$ as used in the monolithic algorithm. This allows us to directly apply the compositional algorithm of Subsection 5.1 to solve Muller games.

Finally, let us mention that the compositional solution of conjunctions of safety conditions was already applied to LTL realizability [18]. Note, however, that although the underlying ideas of our construction in Subsection 5.1 are the same, our technical framework is different.

5.1. Solving Safety Games Compositionally

In this subsection, we describe a general framework to solve games whose winning condition is a conjunction of safety conditions specified by a set of memory structures. The intuition is that a memory structure $\mathfrak{M}$ tracks the play played to far and signals whether the play still satisfies a safety property specified by $\mathfrak{M}$. This way, we can easily combine a conjunction of safety properties into a single safety game by taking the product of the memory structures. When applying this framework to Muller games, we use one memory structure for each set $F \in \mathcal{F}_1$ that keeps track of the scores and accumulators up to the threshold score three.

Given $k > 1$ memory structures $\mathfrak{M}_j = (M_j, \mathrm{Init}_j, \mathrm{Upd}_j)$ for the same arena $\mathcal{A}$, their parallel composition $\mathfrak{M}_1 \times \cdots \times \mathfrak{M}_k = (M_1 \times \cdots \times M_k, \mathrm{Init}, \mathrm{Upd})$ with $\mathrm{Init}(v) = (\mathrm{Init}_1(v), \ldots, \mathrm{Init}_k(v))$ and

$$\mathrm{Upd}((m_1, \ldots, m_k), v) = (\mathrm{Upd}_1(m_1, v), \ldots, \mathrm{Upd}_k(m_k, v))$$

is again a memory structure for $\mathcal{A}$. For $F_j \subseteq M_j$ we define the safety game

$$\mathcal{G} = (\mathcal{A} \times (\mathfrak{M}_1 \times \cdots \times \mathfrak{M}_k), V \times F_1 \times \cdots \times F_k)) \ ,$$

i.e., Player 0 wins if the memory in every structure stays in F_j. The size of $\mathcal{G}$ grows exponentially in k, the number of safety conditions implemented by the memory structures $\mathfrak{M}_j$.

In the following, we show how to determine Player 0's winning region in $\mathcal{G}$ without constructing the complete arena $\mathcal{A} \times (\mathfrak{M}_1 \times \cdots \times \mathfrak{M}_k)$. Let k' be in the range $1 \leq k' < k$ and consider the safety games

- $\mathcal{G}_1 = (\mathcal{A} \times (\mathfrak{M}_1 \times \cdots \times \mathfrak{M}_{k'}), V \times F_1 \times \cdots \times F_{k'})$, and
- $\mathcal{G}_2 = (\mathcal{A} \times (\mathfrak{M}_{k'+1} \times \cdots \times \mathfrak{M}_k), V \times F_{k'+1} \times \cdots \times F_k)$.

We define

$$X = \{(v, m_1, \ldots, m_k) \mid (v, m_1, \ldots, m_{k'}) \in W_0(\mathcal{G}_1) \text{ and } (v, m_{k'+1}, \ldots, m_k) \in W_0(\mathcal{G}_2)\}$$

and the arena $\mathcal{A}_X$ as restriction of $\mathcal{A} \times (\mathfrak{M}_1 \times \cdots \times \mathfrak{M}_k)$ to vertices in X. Note that this arena might have terminal vertices. Denote the set of these vertices by T and consider the safety game $\mathcal{G}_{red} = (\mathcal{A}_X, X \setminus T)$, i.e., Player 0 has to avoid the terminal vertices[6].

The following theorem is proven in [18] in a different technical framework. For the sake of completeness, we present the proof in terms of memory structures here.

Theorem 2. $W_0(\mathcal{G}) = W_0(\mathcal{G}_{red})$.

Proof. Let $(v, m_1, \ldots, m_k) \in W_0(\mathcal{G})$, i.e., Player 0 has a winning strategy σ for $\mathcal{G}$ from v. Towards a contradiction, assume $(v, m_1, \ldots, m_k) \notin W_0(\mathcal{G}_{red})$. We consider two cases.

If $(v, m_1, \ldots, m_k)$ is not even a vertex of $\mathcal{A}_X$, then Player 1 has a positional winning strategy τ from $(v, m_0, \ldots, m_{k'})$ in $\mathcal{G}_1$ or from $(v, m_{k'+1}, \ldots, m_k)$ in $\mathcal{G}_2$. We only consider the first subcase, the second one is analogous. The strategy τ for $\mathcal{G}_1$ can be lifted to a positional strategy τ' for $\mathcal{G}$ via

$$\tau'(v, m_1, \ldots, m_k) = (v', \mathrm{Upd}_1(m_1, v'), \ldots, \mathrm{Upd}_k(m_k, v')) \ ,$$

where v' is the first component of $\tau(v, m_1, \ldots, m_{k'}) = (v', m'_1, \ldots, m'_{k'})$. Note that we have $\mathrm{Upd}_j(m_j, v') = m'_j$ for every $j \leq k'$, i.e., taking the projection to the first k' memory states of a play in $\mathcal{G}$ that is consistent with τ' yields a play in $\mathcal{G}_{red}$ that is consistent with τ. Let ρ be the unique play consistent with σ and τ' that starts in $(v, m_1, \ldots, m_k)$. Since its projection to the first k' memory states is consistent with τ, it reaches a vertex with a memory state in the j-th

[6] So, terminal vertices are unproblematic, as Player 0 loses plays that reach such a vertex.

coordinate that is not in F_j (for some $j \leq k'$). Thus, ρ is not a winning play for Player 0 in $\mathcal{G}$, which yields the desired contradiction.

Now, assume $(v, m_1, \ldots, m_k)$ is a vertex of $\mathcal{A}_X$, but is in $W_1(\mathcal{G}_{red})$. This implies that Player 1 has a strategy τ which ensures that every play that starts in $(v, m_1, \ldots, m_k)$ and is consistent with τ ends up in T. Furthermore, in $\mathcal{G}$ every successor of a vertex in T is not a vertex of $\mathcal{A}_X$. In the previous case, we have argued that Player 1 can win $\mathcal{G}$ from such vertices. Hence, he can do the same from $(v, m_0, \ldots, m_k)$ by first leaving X and then reaching a memory state not in F_j as described above. Hence, $(v, m_1, \ldots, m_k)$ is not in $W_0(\mathcal{G})$, which is again a contradiction.

For the other direction, let $(v, m_1, \ldots, m_k) \in W_0(\mathcal{G}_{red})$, i.e., Player 0 has a strategy σ for $\mathcal{G}_{red}$ that confines every play starting in $(v, m_1, \ldots, m_k)$ to $X \setminus T$. The same strategy is also a strategy for $\mathcal{G}$. Furthermore, every play that is consistent with σ in $\mathcal{G}$ is also consistent with σ in $\mathcal{G}_{red}$. Finally, as X only contains vertices in $V \times F_1 \times \cdots \times F_k$, σ is also a winning strategy for Player 0 in $\mathcal{G}$ from $(v, m_1, \ldots, m_k)$. □

Note that it is necessary to solve $\mathcal{G}_{red}$: it is easy to construct examples where $W_0(\mathcal{G})$ is a strict subset of X, e.g., Player 0 can avoid reaching $M_1 \setminus F_1$ in the first memory structure and can avoid reaching $M_2 \setminus F_2$ in the second memory structure, but not both at the same time.

So, to determine Player 0's winning region in $\mathcal{G}$ it suffices to solve the safety games $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_{red}$. Both $\mathcal{G}_1$ and $\mathcal{G}_2$ are smaller than $\mathcal{G}$, while $\mathcal{G}_{red}$ is at most as large as $\mathcal{G}$, and much smaller if Player 1's winning regions in $\mathcal{G}_1$ and $\mathcal{G}_2$ are large. Furthermore, the games $\mathcal{G}_1$ and $\mathcal{G}_2$ can again be split into smaller games. By inductively splitting the conditions until we have reached safety games with a single memory structure, we can solve $\mathcal{G}$ by solving at most $2k$ smaller safety games: k with a single memory structure, at most $\lceil \frac{k}{2} \rceil$ with (subsets of) two structures, at most $\lceil \frac{k}{4} \rceil$ with (subsets of) three structures, etc., until we are left with one with (subsets of) k structures.

5.2. Solving Muller Games Compositionally

For the remainder of this section, we fix a Muller game $\mathcal{G} = (\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$ with $\mathcal{A} = (V, V_0, V_1, E)$ and $\mathcal{F}_1 = \{F_1, \ldots, F_k\}$. We want to determine the winning regions of $\mathcal{G}$ compositionally using the construction of the previous subsection. To this end, we give a description of the safety game $\mathcal{G}_S$ (as defined in Section 4) for $\mathcal{G}$ in terms of a cartesian product of $\mathcal{A}$ and the parallel composition of memory structures $\mathfrak{M}_j$ for each set F_j.

For every set $F \subseteq V$, there is a deterministic finite automaton $\mathfrak{A}_F$ accepting exactly those $w \in V^+$ with $\mathrm{MaxSc}_{\{F\}}(w) \geq 3$. The automaton uses the state space $\{0, 1, 2\} \times 2^F$ (pairs of a score and an accumulator) with an additional final (sink) state s_F that is reached when the score is incremented to three for the first time, while the transition function implements the definition of Sc_F and Acc_F as described in Section 3.

We turn this automaton into a memory structure $\mathfrak{M}_F = (Q, \mathrm{Init}, \mathrm{Upd})$ with $\mathrm{Init}(v) = \delta(q_0, v)$ and $\mathrm{Upd} = \delta$. Here, Q is the set of states, q_0 is the initial state,

and δ is the transition function of $\mathfrak{A}_F$. The memory is constructed such that we have $\mathrm{Upd}^*(w) = \delta^*(w)$,[7]. Hence, we have $\mathrm{Upd}^*(w) = (\mathrm{Sc}_F(w), \mathrm{Acc}_F(w))$ for every w with $\mathrm{MaxSc}_{\{F\}}(w) \leq 2$, and $\mathrm{Upd}^*(w) = s_F$, if we have $\mathrm{MaxSc}_{\{F\}}(w) > 2$.

For every j let $\mathfrak{M}_j = (M_j, \mathrm{Init}_j, \mathrm{Upd}_j)$ be the memory structure for F_j as described above. Furthermore, let S_j be the set of non-final states of the automaton for F_j. We define $\mathcal{A}_C$ to be $\mathcal{A} \times (\mathfrak{M}_1 \times \cdots \times \mathfrak{M}_k)$ restricted to the vertices reachable from

$$I = \{(v, (\mathrm{Sc}_{F_1}(v), \mathrm{Acc}_{F_1}(v)), \ldots, (\mathrm{Sc}_{F_k}(v), \mathrm{Acc}_{F_k}(v))) \mid v \in V\}$$

after we have deleted all outgoing edges of the vertices that are not in $V \times S_1 \times \cdots \times S_k$. Terminal vertices do not cause problems here since we define them to be unsafe, i.e., as soon as a play reaches such a vertex the winner is certain. Formally, we define the safety game $\mathcal{G}_C = (\mathcal{A}_C, (V \times S_1 \times \cdots \times S_k) \cap V_C\})$, where V_C is the set of vertices of $\mathcal{A}_C$.

We are interested in the vertices in I, since they encode the vertices of the original Muller game $\mathcal{G}$. The following remark states that the winner from these vertices is not changed by deleting the outgoing edges of the unsafe vertices.

Remark 3. *Let $v \in I$. Then, $v \in W_i(\mathcal{G}_C)$ if and only if $v \in W_i(\mathcal{A} \times (\mathfrak{M}_1 \times \cdots \times \mathfrak{M}_k), V \times S_1 \times \cdots \times S_k)$.*

We say that two safety games are isomorphic, if there is a bijection between the sets of states that preserves the partition into the players' positions, the edge relation, and the set of safe states. In this situation, the isomorphism also preserves winning regions. Recall that $\mathcal{G}_S$ is the safety game tracking all scores constructed in the previous section.

Lemma 8. *The safety games $\mathcal{G}_C$ and $\mathcal{G}_S$ are isomorphic.*

Proof. We begin by giving a mapping h from vertices of $\mathcal{G}_S$ to vertices of $\mathcal{G}_C$. For every $w \in \mathrm{Plays}_{<3}$ define

$$h([w]) = (\mathrm{Last}(w), (\mathrm{Sc}_{F_1}(w), \mathrm{Acc}_{F_1}(w)), \ldots, (\mathrm{Sc}_{F_k}(w), \mathrm{Acc}_{F_k}(w))) \ .$$

Recall that s_F is the accepting sink state of the automaton $\mathfrak{A}_F$ that is reached when a score of three is reached for F. Now, let $w \in \mathrm{Plays}_{=3}$ and let j be the unique index such that $\mathrm{Sc}_{F_j}(w) = 3$. It is unique, since no two scores can increase to three at the same time [14, 15]. Furthermore, all other scores of w are at most two. We define

$$\begin{aligned} h([w]) = (\mathrm{Last}(w), &(\mathrm{Sc}_{F_1}(w), \mathrm{Acc}_{F_1}(w)), \ldots, (\mathrm{Sc}_{F_{j-1}}(w), \mathrm{Acc}_{F_{j-1}}(w)), \\ &s_{F_j}, (\mathrm{Sc}_{F_{j+1}}(w), \mathrm{Acc}_{F_{j+1}}(w)), \ldots, (\mathrm{Sc}_{F_k}(w), \mathrm{Acc}_{F_k}(w))) \ . \end{aligned}$$

[7] $\delta^*(w)$ denotes the state reached after processing w in $\mathfrak{A}_F$.

It is easy to prove that the mapping is independent of representatives and therefore well-defined, that it is injective, and that it preserves the partition into the players' positions and the edge relation. The only non-trivial property is surjectivity.

Recall that no two scores can be increased to three at the same time. Hence, no two automata $\mathfrak{A}_F$ can reach their sink state at the same time. Furthermore, we have deleted all outgoing edges of the vertices of $\mathcal{A}_C$ that contain at least one sink state. Thus, every vertex of $\mathcal{A}_C$ has at most one sink state in its components.

First consider vertices of the form $(v, (s_1, A_1), \ldots, (s_k, A_k))$ without a sink state. Since this state is reachable, there is a play prefix w such that $\mathrm{Last}(w) = v$ and $\mathrm{Sc}_{F_j}(w) = s_j$ and $\mathrm{Acc}_{F_j}(w) = A_j$ for every j. Furthermore, w is in $\mathrm{Plays}_{<3}$ since no sink state is reached. Thus, $[w]$ is a vertex of $\mathcal{G}_S$, and it is mapped to $(v, (s_1, A_1), \ldots, (s_k, A_k))$.

Now, consider a vertex of the form $(v, (s_1, A_1), \ldots, s_{F_j}, \ldots, (s_k, A_k))$, i.e., there is a single sink state in the j-th component. Again, as this vertex is reachable, there is a play prefix $w = w_0 \cdots w_n w_{n+1}$ such that $w_{n+1} = v$ and $\mathrm{Sc}_{F_{j'}}(w) = s_{j'}$ and $\mathrm{Acc}_{F_{j'}}(w) = A_{j'}$ for every $j' \neq j$. Furthermore, we have $\mathrm{MaxSc}_{\{F_j\}}(w) = 3$, as the final state s_{F_j} is reached. Since we have deleted all outgoing edges of the unsafe vertices of $\mathcal{A}_C$, we conclude that the sink vertex s_{F_j} is reached by processing w_{n+1}, i.e., the score for F_j is bounded by two in $w_0 \cdots w_n$. Hence, we have $w \in \mathrm{Plays}_{=3}$, which implies that $[w]$ is a vertex of $\mathcal{G}_S$ and is mapped to $(v, (s_1, A_1), \ldots, s_{F_j}, \ldots, (s_k, A_k))$.

Thus, the mapping is also surjective and therefore an isomorphism. □

So, due to Theorem 1 and the fact that the isomorphism preserves winning regions, to determine whether Player 0 wins the Muller game $\mathcal{G}$ from v, it suffices to determine whether Player 0 wins $\mathcal{G}_C$ from

$$h([v]) = (v, (\mathrm{Sc}_{F_1}(v), \mathrm{Acc}_{F_1}(v)), \ldots, (\mathrm{Sc}_{F_k}(v), \mathrm{Acc}_{F_k}(v))) \ .$$

Due to Remark 3, this is equivalent to Player 0 winning $(\mathcal{A} \times (\mathfrak{M}_1 \times \cdots \times \mathfrak{M}_k), V \times S_1 \times \cdots \times S_k)$ from $h([v])$. So, to determine whether v is in the winning region of Player 0 in the Muller game, it suffices the solve a safety game that is induced by a cartesian product of memory structures. Theorem 2 shows that we can do this by splitting the safety game into smaller games and combining their solutions.

To conclude this section, let us briefly discuss the advantages and disadvantages of the compositional algorithm. To this end, consider a Muller game $(\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$. First, we observe that the compositional algorithm is preferable in situations where there are a lot of vertices v_j and small subsets $\mathcal{F}_j \subseteq \mathcal{F}_1$ such that Player 1 wins the Muller game $(\mathcal{A}, \mathcal{F}_0 \cup (\mathcal{F}_1 \setminus \mathcal{F}_j), \mathcal{F}_j)$ from v_j. This implies that Player 1 also wins the original Muller game from v_j and all vertices with v_j in the first component are eliminated from $\mathcal{G}_S$ by the compositional algorithm as soon as all memory structures for the sets $F \in \mathcal{F}_j$ are present in a single subgame. The most extreme example is where each vertex of the Muller game is in V_1, has a self-loop, and each singleton $\{v\}$ is in $\mathcal{F}_1$. Then, solving the

safety games $\mathcal{A} \times \mathfrak{M}_j$ for the singleton sets $F_j = \{v_j\}$ eliminates all vertices of the arena, while the monolithic algorithm from Section 4 constructs the whole safety game $\mathcal{G}_S$.

On the other hand, assume Player 1's winning region in the safety game $\mathcal{G}_S$ contains only the unsafe vertices. Then, no vertex can be eliminated by the compositional algorithm. Thus, the monolithic algorithm is preferable since it only has to solve the game $\mathcal{G}_S$ whereas the compositional algorithm solves many smaller games and also needs to solve the whole game $\mathcal{G}_S$ in the last step.

Furthermore, the performance of the compositional algorithm is influenced by the choice of the splits. For example, consider the situation where Player 1 can enforce a score of three for the same set $F \in \mathcal{F}_1$ from every vertex in V, but Player 0 can prevent a score of three for all other sets in $\mathcal{F}_1$. Then, the best partition of $\mathcal{F}_1$ is $(\{F\}, \mathcal{F}_1 \setminus \{F\})$, which allows us to solve the given game very quickly by considering the set F first. However, every combination of splits where F is considered as one of the last sets makes the algorithm solve many safety games in vain.

6. Permissive Strategies for Muller Games

A parity game $(\mathcal{A}, \Omega)$ consists of an arena $\mathcal{A}$ and a priority function $\Omega \colon V \to \mathbb{N}$. A play ρ is winning for Player 0 if the minimal priority seen infinitely often is even. Bernet et al. introduced permissive strategies for parity games [8], a non-deterministic winning strategy that subsumes the behavior of every positional (non-deterministic) winning strategy. To compute such a strategy, they reduce a parity game to a safety game. The main observation underlying their reduction is the following: if we denote the number of vertices of priority c by n_c, then a (non-deterministic) positional winning strategy for Player 0 does not allow a play in which an odd priority c is visited $n_c + 1$ times without visiting a smaller priority in between. This property can be formulated using scoring functions for parity games as well. The scoring function for a priority c counts the occurrences of c since the last occurrence of a smaller priority (such an occurrence resets the score for c to 0). Hence, our work on Muller games can be seen as a generalization of Bernet et al.'s work. While the bound n_c on the scores in a parity game is straightforward, the bound three for Muller games is far from obvious.

Since both constructions are very similar, it is natural to ask whether we can use the concept of permissive strategies for Muller games. In parity games, we ask for a non-deterministic strategy that subsumes the behavior of every *positional* strategy. As positional strategies do not suffice to win Muller games, we have to give a new definition of permissiveness for such games. In other words, we need to specify the strategies whose behaviors a permissive strategy for a Muller game should subsume. One way to do this is to fix a sufficiently large bound M and to require that a permissive strategy for a Muller game subsumes the behavior of every finite-state winning strategy of size at most M (this was already proposed for parity games in [8]).

However, we prefer to take a different approach. By closely inspecting the reduction of parity to safety games, it becomes apparent that the induced strategy does not only subsume the behavior of every positional winning strategy, but rather the behavior of every strategy that prevents the opponent from reaching a score of $n_c + 1$ for some odd priority c (in terms of scoring functions for parity games). It is this formulation that we extend to Muller games: a (non-deterministic) winning strategy for a Muller game is permissive, if it subsumes the behavior of every (non-deterministic) winning strategy that prevents the losing player from reaching a score of three. We formalize this notion in the following and show how to compute such strategies from the safety game constructed in the previous section.

We begin by introducing some additional notation. A multi-strategy for Player i in (V, V_0, V_1, E) is a mapping $\sigma\colon V^*V_i \to 2^V \setminus \{\emptyset\}$ such that $v' \in \sigma(wv)$ implies $(v, v') \in E$. A play ρ is consistent with σ if $\rho_{n+1} \in \sigma(\rho_0 \cdots \rho_n)$ for every n with $\rho_n \in V_i$. We still denote the plays starting in a vertex v that are consistent with σ by $\mathrm{Beh}(v, \sigma)$ and define $\mathrm{Beh}(W, \sigma) = \bigcup_{v \in W} \mathrm{Beh}(v, \sigma)$ for every subset $W \subseteq V$. A multi-strategy σ is winning for Player 0 from a set of vertices W in a game $(\mathcal{A}, \mathrm{Win})$ if $\mathrm{Beh}(W, \sigma) \subseteq \mathrm{Win}$, and a multi-strategy τ is winning for Player 1 from W, if $\mathrm{Beh}(W, \tau) \subseteq V^\omega \setminus \mathrm{Win}$. It is clear that the winning regions of a game do not change when we allow multi-strategies instead of standard strategies.

To define finite-state multi-strategies we have to allow a next-move function to return more than one vertex, i.e., we have $\mathrm{Nxt}\colon V_i \times M \to 2^V \setminus \{\emptyset\}$ such that $v' \in \mathrm{Nxt}(v, m)$ implies $(v, v') \in E$. A memory structure $\mathfrak{M}$ and Nxt implement a multi-strategy σ via $\sigma(wv) = \mathrm{Nxt}(v, \mathrm{Upd}^*(wv))$.

Definition 3. *A multi-strategy σ' for a Muller game $\mathcal{G}$ is permissive, if*

1. *σ' is a winning strategy from every vertex in $W_0(\mathcal{G})$, and*
2. *$\mathrm{Beh}(v, \sigma) \subseteq \mathrm{Beh}(v, \sigma')$ for every multi-strategy σ and every vertex v with $\mathrm{MaxSc}_{\mathcal{F}_1}(\rho) \leq 2$ for every $\rho \in \mathrm{Beh}(v, \sigma)$.*

The original definition for parity games replaces the second condition by the following requirement: $\mathrm{Beh}(v, \sigma) \subseteq \mathrm{Beh}(v, \sigma')$ for every positional multi-strategy σ and every v from which σ is winning.

Example 4. *Once again consider the Muller game of Example 2. Starting at vertex 1, moving to 0 is consistent with a winning strategy for Player 0 that bounds Player 1's scores by two. Similarly, moving to 2 is also consistent with a winning strategy for Player 0 that bounds Player 1's scores. Hence, we have $\sigma'(1) = \{0, 2\}$ for every permissive strategy σ'. Now consider the play prefix 10. Here it is Player 1's turn and he can use the self-loop either infinitely often (which yields a play that is winning for Player 0) or only finitely often (say n times) before moving back to vertex 1. In this situation, i.e., with play prefix $10^{n+1}1$, a strategy that bounds Player 1's scores by two has to move to vertex 2. Hence, we must have $\sigma'(10^{n+1}1) \supseteq \{2\}$. However, it is possible that we also have $1 \in \sigma'(10^{n+1}1)$, since a permissive strategy may allow more plays than the*

ones of strategies that bound Player 1*'s scores by two. However, at some point,* σ' *has to disallow the move back to vertex* 0*, otherwise it would allow a play that is losing for her.*

Using the safety game $\mathcal{G}_S$ from Section 4 we prove that Player 0 always has a finite-state permissive strategy and show how to compute one.

Theorem 3. *Let* $\mathcal{G}$ *be a Muller game and* $\mathcal{G}_S$ *the corresponding safety game as above. Then, Player* 0 *has a finite-state permissive strategy for* $\mathcal{G}$ *with memory states* $W_0(\mathcal{G}_S)$*.*

The proof is very similar to the one for Theorem 1.2 (cf. the construction in the proof of Lemma 5), but we have to use all vertices in $W_0(\mathcal{G}_S)$ as memory states to implement a permissive strategy, only using the maximal ones (restricted to those reachable by some fixed winning strategy for the safety games) does not suffice. Furthermore, the next-move function does not return one successor that guarantees a memory update to a state from $W_0(\mathcal{G}_S)$, but it returns all such states.

Proof. We define $\mathfrak{M} = (M, \text{Init}, \text{Upd})$ where $M = W_0(\mathcal{G}_S) \cup \{\bot\}$ [8],

$$\text{Init}(v) = \begin{cases} [v] & \text{if } [v] \in W_0(\mathcal{G}_S), \\ \bot & \text{otherwise,} \end{cases}$$

and

$$\text{Upd}([w], v) = \begin{cases} [wv] & \text{if } [wv] \in W_0(\mathcal{G}_S), \\ \bot & \text{otherwise.} \end{cases}$$

Hence, we have $\text{Upd}^*(w) = [w] \in W_0(\mathcal{G}_S)$ as long as every prefix x of w satisfies $[x] \in W_0(\mathcal{G}_S)$, and $\text{Upd}^*(w) = \bot$ otherwise. We define Nxt by $\text{Nxt}(v, \bot) = \{v'\}$ for some successor v' of v and

$$\text{Nxt}(v, [w]) = \begin{cases} \{v' \mid [wv'] \in W_0(\mathcal{G}_S)\} & \text{if } [w] \in W_0(\mathcal{G}_S) \text{ and } \text{Last}(w) = v, \\ \{v''\} & \text{otherwise, for some successor } v'' \text{ of } v. \end{cases}$$

Since every vertex in $W_0(\mathcal{G}_S) \cap V_0^S$ has at least one successor in $W_0(\mathcal{G}_S)$, the next-move function always returns a non-empty set of successors of v in $\mathcal{G}$.

It remains to show that the strategy σ' implemented by $\mathfrak{M}$ and Nxt is permissive. We begin by showing that σ' is winning from every vertex $v \in W_0(\mathcal{G})$: due to Lemma 4, we have $[v] \in W_0(\mathcal{G}_S)$. Hence, the memory is initialized with $[v] \in W_0(\mathcal{G}_S)$. A simple induction shows $\text{Upd}^*(w) = [w] \in W_0(\mathcal{G}_S)$ for every play prefix that starts in $[v]$ is consistent with σ'. This bounds Player 1's scores by two. Hence, σ' is indeed winning from v.

[8] Again, we use the memory state $\bot$ to simplify our proof. It is not reachable via plays that are consistent with the strategy implemented by $\mathfrak{M}$ and can therefore eliminated and its incoming transitions can be redefined arbitrarily.

Finally, consider a multi-strategy σ and a vertex v with $\mathrm{MaxSc}_{\mathcal{F}_1}(\rho) \leq 2$ for every play $\rho \in \mathrm{Beh}(v, \sigma)$. We have to show that every play $\rho \in \mathrm{Beh}(v, \sigma)$ is consistent with σ'. Since σ is winning from v (as it bounds Player 1's scores), we have $v \in W_0(\mathcal{G})$. Now, assume ρ is not consistent with σ' and let $\rho_0 \cdots \rho_n \rho_{n+1}$ be the shortest prefix such that $\rho_{n+1} \notin \sigma'(\rho_0 \cdots \rho_n)$. Then, we have $[\rho_0 \cdots \rho_n \rho_{n+1}] \notin W_0(\mathcal{G}_S)$. Hence, Player 1 has a strategy to enforce a visit to $V^S \setminus F$ in $\mathcal{G}_S$ starting in $[\rho_0 \cdots \rho_n \rho_{n+1}]$. Player 1 can mimic this strategy in $\mathcal{G}$ to enforce a score of three against every strategy of Player 0 when starting with the play prefix $\rho_0 \cdots \rho_n \rho_{n+1}$. Since this prefix is consistent with σ, which we have assumed to bound Player 1's scores by two, we have derived the desired contradiction. □

7. Safety Reductions for Infinite Games

Recall that the classical notion of a game reduction (as defined in Section 2) requires the extended play ρ' to be winning for the same player that wins the original play ρ. It is well known that there are no such classical game reductions from Muller games to safety games. The reason is that a reduction induces a continuous function mapping (winning) plays of the original game to (winning) plays of the reduced game. The existence of such functions is tied to topological properties of the sets of winning plays in both games, which are not satisfied in the case of Muller and safety games. However, we transformed a Muller game to a safety game which allowed us to determine the winning regions and a winning strategy for one player. This is possible, since our reduction does not induce a continuous function: a play is stopped as soon as Player 1 reaches a score of three. As it can (in general) be extended to be winning for both players, i.e., also for Player 1, this violates the requirement on a classical reduction.

In this section, we briefly discuss the reason why Muller games can not be reduced to safety games using classical reductions, and then we present a novel type of game reduction that allows us to reduce many games known from the literature to safety games. The advantage of this safety reduction is that the reduced game is always a safety game. Hence, we can determine the winning regions of various games from different levels of the Borel hierarchy using the same technique. However, we only obtain a winning strategy for one player, and we need some information on the type of winning strategies a player has in such a game.

Formally, the set $\mathrm{Win}_M \subseteq V^\omega$ of winning plays of a Muller game is in general on a higher level of the Borel hierarchy than the set $\mathrm{Win}_S \subseteq U^\omega$ of winning plays of a safety game. Hence, in general, there exists no continuous (in the Cantor topology) function $f\colon V^\omega \to U^\omega$ such that $\rho \in \mathrm{Win}_M$ if and only if $f(\rho) \in \mathrm{Win}_S$ (see, e.g., [21]). Since the mapping from a play in $\mathcal{A}$ to its extended play in $\mathcal{A} \times \mathfrak{M}$ is continuous, we obtain the following negative result (which holds for other pairs of games as well).

Corollary 2. *In general, Muller games cannot be reduced to safety games.*

To overcome this, we present a new type of game reduction which encompasses the construction presented above and applies to many other games.

Definition 4. *A game $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ with vertex set V is (finite-state) safety reducible, if there is a regular language $L \subseteq V^*$ of finite words such that:*

- *For every play $\rho \in V^\omega$: if $\mathrm{Pref}(\rho) \subseteq L$, then $\rho \in \mathrm{Win}$.*
- *If $v \in W_0(\mathcal{G})$, then Player 0 has a strategy σ with $\mathrm{Pref}(\mathrm{Beh}(v, \sigma)) \subseteq L$.*

Note that a strategy σ satisfying $\mathrm{Pref}(\mathrm{Beh}(v, \sigma)) \subseteq L$ is winning for Player 0 from v. Many games appearing in the literature on infinite games are safety reducible, although these reductions neither yield fast solution algorithms nor optimal memory structures. However, all the winning conditions presented below are on a higher level of the Borel hierarchy than safety conditions, i.e., all these games can not be (classically) reduced to safety games. For definitions see [22] or the papers cited below.

- In a Büchi game $\mathcal{G}$, Player 0 has a positional winning strategy. Every play consistent with this strategy visits a vertex in F at least every $k = |V \setminus F|$ steps. Hence, $\mathcal{G}$ is safety reducible with $L = \mathrm{Pref}(((V \setminus F)^{\leq k} \cdot F)^\omega)$.
- In a co-Büchi game $\mathcal{G}$, Player 0 has a positional winning strategy. Every play consistent with this strategy stays in F after visiting each vertex in $V \setminus F$ at most once. Hence, $\mathcal{G}$ is safety reducible with $L = \mathrm{Pref}(\{w \cdot F^\omega \mid$ each $v \in V \setminus F$ appears at most once in $w\})$.
- In a request-response game $\mathcal{G}$, Player 0 has a finite-state winning strategy such that in every consistent play every request is answered within $k = |V| \cdot r \cdot 2^{r+1}$ steps, where r is the number of request-response pairs [23]. Hence, $\mathcal{G}$ is safety reducible to the language of prefixes of plays in which every request is answered within k steps.
- In a parity game, Player 0 has a positional winning strategy. Every play consistent with this strategy does not visit $n_c + 1$ vertices with an odd priority c without visiting a smaller priority in between, where n_c is the number of vertices with priority c. Hence, $\mathcal{G}$ is safety reducible to the language of prefixes of plays satisfying this condition for every odd c.
- The results of Piterman and Pnueli on progress-measures for Rabin and Streett games [24] can also be rephrased in terms of safety reductions.
- Energy progress measures for solving energy (and mean-payoff) games can also be used to give a safety reduction for such games, as described in [25].
- Lemma 3 shows that a Muller game is safety reducible to the language of prefixes of plays that never reach a score of three for Player 1.

- In a finitary parity game $\mathcal{G}$, Player 0 has a positional winning strategy [10]. Such a strategy guarantees that all but $|V|$ vertices of odd color are followed by a vertex of smaller even color within $|V|$ steps. Hence, $\mathcal{G}$ is reducible to the language of play prefixes satisfying this condition. The same is true for finitary Streett games (albeit with larger bounds) and parity and Streett games with costs [11].

Let $\mathcal{G}$ be a game with arena $\mathcal{A} = (V, V_0, V_1, E)$ and let $\mathfrak{A} = (Q, V, q_0, \delta, F)$ be a deterministic finite automaton (DFA) recognizing a language over V. As in Section 5, we turn $\mathfrak{A}$ into a memory structure $\mathfrak{M} = (Q, \mathrm{Init}, \mathrm{Upd})$ with $\mathrm{Init}(v) = \delta(q_0, v)$ and $\mathrm{Upd} = \delta$.

Theorem 4. *Let $\mathcal{G}$, $\mathfrak{A}$, and $\mathfrak{M}$ be as above such that $\mathcal{G}$ is safety reducible with language $L(\mathfrak{A})$. Define the safety game $\mathcal{G}' = (\mathcal{A} \times \mathfrak{M}, V \times F)$.*

1. *Let $v \in V$. Then, $v \in W_0(\mathcal{G})$ if and only if $(v, \mathrm{Init}(v)) \in W_0(\mathcal{G}')$.*
2. *Player 0 has a uniform finite-state winning strategy for $\mathcal{G}$ with memory Q.*

Proof. 1. Let $v \in W_0(\mathcal{G})$ and let σ be a winning strategy for Player 0 from v that satisfies $\mathrm{Pref}(\mathrm{Beh}(v, \sigma)) \subseteq L(\mathfrak{A})$. We define $\sigma'((v_1, m_1) \cdots (v_n, m_n)) = (v', \mathrm{Upd}(m_n, v'))$ where $v' = \sigma(v_1 \cdots v_n)$. We show that σ' is winning for Player 0 from $(v, \mathrm{Init}(v))$ in $\mathcal{G}'$. A simple induction shows that $(v_1, m_1) \cdots (v_n, m_n)$ being consistent with σ' implies $v_1 \cdots v_n$ being consistent with σ.

So, suppose σ' is not winning in the safety game, i.e., there exists a play prefix $w' = (v_1, m_1) \cdots (v_n, m_n)$ in $\mathcal{G}'$ starting in $(v, \mathrm{Init}(v))$ that is consistent with σ such that $(v_n, m_n) \notin V \times F$. Let $w = v_1 \cdots v_n$. Since $q_0 m_1 \cdots m_n$ is the run of $\mathfrak{A}$ on w, we have $w \notin L(\mathfrak{A})$. Furthermore, as w' is consistent with σ', w is consistent with σ. This contradicts our assumption on σ allowing only play prefixes that are in $L(\mathfrak{A})$.

For the other direction, we use $\mathfrak{M}$ to implement a uniform finite-state winning strategy for $\mathcal{G}$. Fix a uniform positional winning strategy σ' for Player 0 for $\mathcal{G}'$ and define Nxt by $\mathrm{Nxt}(v, m) = v'$, if $\sigma'(v, m) = (v', m')$ for some m'. Let $(v, \mathrm{Init}(v)) \in W_0(\mathcal{G}')$. We show that the strategy σ induced by $\mathfrak{M}$ and Nxt is winning for Player 0 from v. If $v_1 \cdots v_n$ starts in v and is consistent with σ, then the extended play $(v_1, m_1) \cdots (v_n, m_n)$ of w (where $m_j = \mathrm{Upd}^*(v_1 \cdots v_j)$) starts in $(v, \mathrm{Init}(v))$ and a simple induction shows that it is consistent with σ'. Hence, we have $\mathrm{Pref}(\mathrm{Beh}(v, \sigma)) \subseteq L(\mathfrak{A})$, as the memory simulates the run of $\mathfrak{A}$ on w and does not leave F. Hence, σ is indeed a winning strategy from v.

2. We have $\{(v, \mathrm{Init}(v)) \mid v \in W_0(\mathcal{G})\} \subseteq W_0(\mathcal{G}')$ due to the first part of the proof for Statement 1 of this theorem. Hence, the construction in the second part of the proof yields a uniform finite-state winning strategy for Player 0 with memory states Q. □

It is easy to show that every game in which Player 0 has a finite-state winning strategy is safety-reducible to the prefixes of plays consistent with this strategy, which is a regular language. However, this construction requires a finite-state winning strategy to begin with, i.e., there is no need for a safety reduction.

If $\mathcal{G}$ is determined, then Theorem 4.1 is equivalent to $v \in W_i(\mathcal{G})$ if and only if $(v, \mathrm{Init}(v)) \in W_i(\mathcal{G}')$. Hence, all games discussed above can be solved by solving safety games. Furthermore, the new notion of reduction allows us to generalize permissiveness to all games discussed in Example 3: if a game is safety reducible to L, then we can construct a multi-strategy that allows every play ρ in which Player 1 cannot leave L starting from any prefix of ρ. Thereby, we obtain what one could call L-permissive strategies. For example, this allows us to construct the most general non-deterministic winning strategy in a request-response game that guarantees a fixed bound on the waiting times.

Let us conclude by mentioning that safety reducibility of parity games was used to construct an algorithm for parity games [16] and to compute permissive strategies for parity games [8] while safety reducibility of energy games was used to design an algorithm for solving energy and mean-payoff games [25]. Furthermore, the safety reducibility of co-Büchi games is used in work on bounded synthesis [17] and LTL realizability [18], so-called "Safraless" constructions [19] which do not rely on determinization of automata.

8. Conclusion

We have shown how to translate a Muller game into a safety game to determine both winning regions and a finite-state winning strategy for one player. Then, we generalized this construction to a new type of reduction from infinite games to safety games with the same properties. We exhibited several implicit applications of such reductions in the literature as well as several new ones. Our reduction from Muller games to safety games is implemented in the tool `GAVS+`[9] [26].

Our construction is based on the notion of scoring functions for Muller games. Considering the maximal score the opponent can achieve against a strategy leads to a hierarchy of all finite-state strategies for a given game. Previous work has shown that the third level of this hierarchy is always non-empty, and there are games in which the second level is empty. Currently, there is no non-trivial characterization of the games whose first or second level of the hierarchy is non-empty, respectively.

The quality of a strategy can be measured by its level in the hierarchy. We conjecture that there is always a finite-state winning strategy of minimal size in the least non-empty level of this hierarchy, i.e., there is no tradeoff between size and quality of a strategy. This tradeoff may arise in many other games for which a quality measure is defined. Also, a positive resolution of the conjecture would decrease the search space for a smallest finite-state strategy.

Furthermore, we used scores to construct a novel antichain-based memory structure for Muller games. The antichain is induced by a winning strategy for the safety game. It is open how the choice of such a strategy influences the size of the memory structure and how heuristic approaches to computing winning

[9] See `http://www6.in.tum.de/~chengch/gavs/` for details and to download the tool.

strategies that only visit a small part of the arena [27] influence the performance of our reduction.

Finally, tracking score values allows us to develop a compositional algorithm for solving Muller games. Rather than solving a large safety game as the monolithic algorithm above, our compositional algorithm solves several (smaller) safety games. An investigation of classes of Muller games on which the compositional method runs faster or needs less memory than the monolithic algorithm is a part of further research. It is also open whether there are classes of Muller games for which the anti-chain based memory structure is smaller than the LAR memory and the one induced by Zielonka trees.

Furthermore, there is a tight connection between permissive strategies, progress measure algorithms, and safety reductions for parity games: the progress measure algorithm due to Jurdziński [16] and the reduction from parity games to safety game due to Bernet et al. [8] to compute permissive strategies are essentially the same. Whether the safety reducibility of Muller games can be turned into a progress measure algorithm is subject to ongoing research.

Finally, strategies that bound the losing player's scores exist even in games in infinite arenas [28]. It remains to be investigated whether the results proven for Muller games in the present paper can be lifted to infinite games in infinite arenas.

Acknowledgments. The authors want to thank Wolfgang Thomas for bringing McNaughton's work to their attention, Wladimir Fridman for fruitful discussions, Chih-Hong Cheng for his implementation of the algorithm, and Jean-François Raskin for suggesting to investigate the compositional solution of Muller games.

References

[1] W. Zielonka, Infinite games on finitely coloured graphs with applications to automata on infinite trees, Theor. Comput. Sci. 200 (1998) 135–183.

[2] R. McNaughton, Infinite games played on finite graphs, Ann. Pure Appl. Logic 65 (1993) 149–184.

[3] F. Horn, Explicit Muller games are PTIME, in: R. Hariharan, M. Mukund, V. Vinay (Eds.), FSTTCS, volume 2 of *LIPIcs*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2008, pp. 235–243.

[4] S. Dziembowski, M. Jurdziński, I. Walukiewicz, How much memory is needed to win infinite games?, in: LICS, IEEE Computer Society, 1997, pp. 99–110.

[5] Y. Gurevich, L. Harrington, Trees, automata, and games, in: STOC, ACM, 1982, pp. 60–65.

[6] F. Horn, W. Thomas, N. Wallmeier, Optimal strategy synthesis in request-response games, in: S. D. Cha, J.-Y. Choi, M. Kim, I. Ldoi,

M. Viswanathan (Eds.), ATVA, volume 5311 of *LNCS*, Springer, 2008, pp. 361–373.

[7] M. Zimmermann, Time-optimal winning strategies for poset games, in: S. Maneth (Ed.), CIAA, volume 5642 of *LNCS*, Springer, 2009, pp. 217–226.

[8] J. Bernet, D. Janin, I. Walukiewicz, Permissive strategies: from parity games to safety games, ITA 36 (2002) 261–275.

[9] P. Bouyer, N. Markey, J. Olschewski, M. Ummels, Measuring permissiveness in parity games: Mean-payoff parity games revisited, in: [29], pp. 135–149.

[10] K. Chatterjee, T. A. Henzinger, F. Horn, Finitary winning in ω-regular games, ACM Trans. Comput. Log. 11 (2009).

[11] N. Fijalkow, M. Zimmermann, Cost-Parity and Cost-Streett Games, in: D. D'Souza, T. Kavitha, J. Radhakrishnan (Eds.), FSTTCS, volume 18 of *LIPIcs*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012, pp. 124–135.

[12] R. Bloem, K. Chatterjee, T. A. Henzinger, B. Jobstmann, Better quality in synthesis through quantitative objectives, in: A. Bouajjani, O. Maler (Eds.), CAV, volume 5643 of *LNCS*, Springer, 2009, pp. 140–156.

[13] P. Černý, K. Chatterjee, T. A. Henzinger, A. Radhakrishna, R. Singh, Quantitative synthesis for concurrent programs, in: G. Gopalakrishnan, S. Qadeer (Eds.), CAV, volume 6806 of *LNCS*, Springer, 2011, pp. 243–259.

[14] R. McNaughton, Playing infinite games in finite time, in: A. Salomaa, D. Wood, S. Yu (Eds.), A Half-Century of Automata Theory, World Scientific, 2000, pp. 73–91.

[15] J. Fearnley, M. Zimmermann, Playing Muller games in a hurry, Int. J. Found. Comput. Sci. 23 (2012) 649–668.

[16] M. Jurdziński, Small progress measures for solving parity games, in: H. Reichel, S. Tison (Eds.), STACS, volume 1770 of *LNCS*, Springer, 2000, pp. 290–301.

[17] S. Schewe, B. Finkbeiner, Bounded synthesis, in: K. S. Namjoshi, T. Yoneda, T. Higashino, Y. Okamura (Eds.), ATVA, volume 4762 of *LNCS*, Springer, 2007, pp. 474–488.

[18] E. Filiot, N. Jin, J.-F. Raskin, Antichains and compositional algorithms for LTL synthesis, Form. Method. Syst. Des. 39 (2011) 261–296.

[19] O. Kupferman, M. Y. Vardi, Safraless decision procedures, in: FOCS, IEEE Computer Society, 2005, pp. 531–542.

[20] R. Ehlers, Small witnesses, accepting lassos and winning strategies in omega-automata and games, CoRR abs/1108.0315 (2011).

[21] A. Kechris, Classical Descriptive Set Theory, volume 156 of *Graduate Texts in Mathematics*, Springer, 1995.

[22] E. Grädel, W. Thomas, T. Wilke (Eds.), Automata, Logics, and Infinite Games: A Guide to Current Research, volume 2500 of *LNCS*, Springer, 2002.

[23] N. Wallmeier, P. Hütten, W. Thomas, Symbolic synthesis of finite-state controllers for request-response specifications, in: O. H. Ibarra, Z. Dang (Eds.), CIAA, volume 2759 of *LNCS*, Springer, 2003, pp. 11–22.

[24] N. Piterman, A. Pnueli, Faster solutions of Rabin and Streett games, in: LICS, IEEE Computer Society, 2006, pp. 275–284.

[25] L. Brim, J. Chaloupka, L. Doyen, R. Gentilini, J.-F. Raskin, Faster algorithms for mean-payoff games, Form. Method. Syst. Des. 38 (2011) 97–118.

[26] C.-H. Cheng, A. Knoll, M. Luttenberger, C. Buckl, GAVS+: An open platform for the research of algorithmic game solving, in: P. A. Abdulla, K. R. M. Leino (Eds.), TACAS, volume 6605 of *LNCS*, Springer, 2011, pp. 258–261.

[27] D. Neider, Small strategies for safety games, in: [29], pp. 306–320.

[28] W. Fridman, M. Zimmermann, Playing pushdown parity games in a hurry, in: M. Faella, A. Murano (Eds.), GandALF, volume 96 of *EPTCS*, pp. 183–196.

[29] T. Bultan, P.-A. Hsiung (Eds.), Automated Technology for Verification and Analysis, 9th International Symposium, ATVA 2011, Taipei, Taiwan, October 11-14, 2011. Proceedings, volume 6996 of *LNCS*, Springer, 2011.